EXAMINER'S AMENDMENT

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Mr. Jacob Rohwer (Reg. No. 61,229) on 5/7/09.

1. In the claims:

Rewrite the claims as follows:

a. Claim 1;

A software architecture implemented at least in part by a computing device for a distributed computing system comprising:

a plurality of applications configured to handle requests submitted by remote devices over a network, the plurality of applications being written in different programming languages;

an application program interface to present functions used by the plurality of applications to access network and computing resources of the distributed computing system, wherein the application program interface comprises:

a first group of services related to creating Web applications, the first group of services comprising:

constructing Web services;

temporary caching resources;

performing an initial configuration;

creating controls and Web pages;

enabling security in Web server applications; and

accessing session state values;

a second group of services related to constructing client applications, the second group of services comprising:

creating windowing graphical user interface environments; and

enabling graphical functionality;

a third group of services related to data and handling XML documents, the third group of services comprising:

enabling management of data from multiple data sources; and

functions that enable XML processing; and

a fourth group of services related to base class libraries, the fourth group of services comprising:

defining various collections of objects;

accessing configuration settings and handling errors in configuration files;

debugging and tracing code execution;

customizing data according to cultural related information;

inputting and outputting of data;

enabling a programming interface to network protocols;

viewing loaded types, methods, and fields;

creating, storing and managing various culture-specific resources;

enabling system security and permissions;

installing and running services;

enabling character encoding;

enabling multi-threaded programming; and

facilitating runtime operations; and

a common language runtime layer that translates the plurality of applications written in different programming languages into an intermediate language, the intermediate language being:

executed natively by the common language runtime layer; and

configured to access the network and computing resources or the first, second, third, and fourth group of services requested by the remote devices, whereby a seamless integration between multi-language application development, with cross language inheritance is allowed and a robust and secure execution environment for multiple programming languages is provided.

b. Claim 4 (Canceled)

c. Claim 5;

An application program interface embodied on one or more computer readable storage media, comprising:

a first group of services related to creating Web applications, the first group of services comprising:

constructing Web services;

temporary caching resources;

performing an initial configuration;

creating controls and Web pages;

enabling security in Web server applications; and

accessing session state values;

a second group of services related to constructing client applications, the second group of services comprising:

creating windowing graphical user interface environments; and

enabling graphical functionality;

a third group of services related to data and handling XML documents, the third group of services comprising:

enabling management of data from multiple data sources; and

functions that enable XML processing; and

a fourth group of services related to base class libraries, the fourth group of services comprising:

defining various collections of objects;

accessing configuration settings and handling errors in configuration files;

debugging and tracing code execution;

customizing data according to cultural related information;

inputting and outputting of data;

enabling a programming interface to network protocols;

viewing loaded types, methods, and fields;

creating, storing and managing various culture-specific resources;

enabling system security and permissions;

installing and running services;

enabling character encoding;

enabling multi-threaded programming; and

facilitating runtime operations; and

a common language runtime layer that translates Web applications written in different programming languages into an intermediate language, the intermediate language being:

executed natively by the common language runtime layer; and

configured to access resources or the first, second, third, and fourth group of services, whereby a seamless integration between multi-language application development, with cross language inheritance is allowed and a robust and secure execution environment for multiple programming languages is provided, wherein the seamless integration allows for the ability to use a particular code module written in a first programming language with a code module written in a second programming language.

d. Claims 6-9 (Cancelled)

e. Claim 11;

A distributed computer software architecture implemented at least in part by a computing device, comprising:

one or more applications written in different programming languages and configured to be executed on one or more computing devices, the one or more applications written in different programming languages handling requests submitted from remote computing devices;

a networking platform to support the one or more applications;

an application programming interface to interface the one or more applications with the networking platform, wherein the application program interface comprises:

a first group of services related to creating Web applications, the first group of services comprising:

constructing Web services;

temporary caching resources;

performing an initial configuration;

creating controls and Web pages;

enabling security in Web server applications; and

accessing session state values;

a second group of services related to constructing client applications, the second group of services comprising:

creating windowing graphical user interface environments; and

enabling graphical functionality;

a third group of services related to data and handling XML documents, the third group of services comprising:

enabling management of data from multiple data sources; and

functions that enable XML processing; and

a fourth group of services related to base class libraries, the fourth group of services comprising:

defining various collections of objects;

accessing configuration settings and handling errors in configuration files;

debugging and tracing code execution;

customizing data according to cultural related information;

inputting and outputting of data;

enabling a programming interface to network protocols;

viewing loaded types, methods, and fields;

creating, storing and managing various culture-specific resources;

enabling system security and permissions;

installing and running services;

enabling character encoding;

enabling multi-threaded programming; and

facilitating runtime operations; and

a common language runtime layer that translates the one or more applications written in different programming languages into an intermediate language being executed natively by the common runtime layer and configured to access resources or the first, second, third, and fourth group of services requested by the remote devices, whereby a seamless integration, with cross-language inheritance, between the one or more applications developed with multiple programming languages and the computing device is provided.

f. Claims 13-17 (Cancelled)

g. Claim 18;

A computer system comprising:

one or more microprocessors; and

one or more software programs that are written in different programming languages and utilize an application program interface to request services from an operating system through a common language runtime layer, the application program interface comprising:

separate commands to request services consisting of the following groups of services:

A. a first group of services related to creating Web applications, the first group of services comprising:

constructing Web services;

temporary caching resources;

performing an initial configuration;

creating controls and Web pages;

enabling security in Web server applications; and

accessing session state values;

B. a second group of services related to constructing client applications, the second group of services comprising:

creating windowing graphical user interface environments; and

enabling graphical functionality;

C. a third group of services related to data and handling XML documents, the third group of services comprising:

enabling management of data from multiple data sources; and

functions that enable XML processing;

D. a fourth group of services related to base class libraries, the fourth group of services comprising:

defining various collections of objects;

accessing configuration settings and handling errors in configuration files;

debugging and tracing code execution;

customizing data according to cultural related information;

inputting and outputting of data;

enabling a programming interface to network protocols;

viewing loaded types, methods, and fields;

creating, storing and managing various culture-specific resources;

enabling system security and permissions;

installing and running services;

enabling character encoding;

enabling multi-threaded programming; and

facilitating runtime operations; and

a common language runtime layer that allows seamless multi-language development, with cross language inheritance and translates the one or more software

programs written in different programming languages into an intermediate language, wherein the intermediate language is executed natively by the common language runtime layer and is configured to access the first, second, third, and fourth group of services requested by the one or more software programs.

h. Claims 19-25 (Cancelled)

i. Claim 26;

A method implemented at least in part by a computer, the method comprising:

creating a first namespace with functions that enable browser/server communication, the first namespace defining classes that facilitate:

construction and use of Web services;

temporary caching of resources;

an initial configuration;

creation of controls and Web pages;

security in Web server applications; and

access to session state values;

creating a second namespace with functions that enable drawing and

construction of client applications, the second namespace defining classes that facilitate:

creation of windowing graphical user interface environments; and

graphical functionality;

creating a third namespace with functions that enable connectivity to data sources and XML functionality, the third namespace defining classes that facilitate:

management of data from multiple data sources; and

processing of XML documents;

creating a fourth namespace with functions that enable system and runtime functionality, the fourth namespace defining classes that facilitate:

programmatic access to configuration settings and handling of errors in configuration files;

application debugging and code execution tracing;

customization of data according to cultural related information;

inputting and outputting of data;

interfacing to network protocols;

viewing loaded types, methods, and fields;

creation, storage and management of various culture-specific resources;

system security and permissions;

installation and running of services;

character encoding;

multi-threaded programming; and

runtime operations; and

providing a common language runtime layer that translates Web applications written in different programming languages into an intermediate language, the intermediate language being:

executed natively by the common language runtime layer; and

configured to access resources or services requested by the client applications, whereby a seamless integration between multi-language application development, with cross-language inheritance is allowed and a robust and secure execution environment for multiple programming languages is provided.

j. Claims 27-30 (Cancelled)

k. Claim 31;

A method implemented at least in part by a computer, the method comprising: calling one or more first functions to facilitate browser/server communication, the first functions comprising functions for construction and use of Web services, temporary caching of resources, an initial configuration, creation of controls and pages that will appear as user interfaces, securing Web server applications and accessing session state values;

calling one or more second functions to facilitate construction of client applications, the second functions comprising functions for creating window graphical user interface environments and graphical functionality;

calling one or more third functions to facilitate connectivity to data sources and XML functionality, the third functions comprising functions for management of data from multiple data sources and XML processing;

calling one or more fourth functions to access system and runtime resources, the fourth functions comprising functions for programmatic access to configuration settings, application debugging and code execution tracing, customization of text according to cultural related information, synchronous and asynchronous reading from and writing to data streams and files, creation and management of various culture-specific resources, system security and permissions, installation and running of services, character encoding, and multi-threaded programming; and

using a common language runtime layer that translates Web applications written in different programming languages into an intermediate language that is:

executed natively by the common language runtime layer; and

configured to access resources or services requested by the client applications, whereby a seamless integration between multi-language application development, with cross-language inheritance is allowed and a robust and secure execution environment for multiple programming languages is provided, wherein the seamless integration allows for the ability to use a particular code module written in a first programming language with a code module written in a second programming language.

I. Claims 32-35 (Canceled)

m. Claim 36;

A method implemented at least in part by a computer, the method comprising: receiving one or more calls to one or more first functions to facilitate browser/server communication, the one or more first functions comprising functions for construction and use of Web services, temporary caching of resources, initial configuration, creation of controls and pages that will appear as user interfaces, securing Web server applications and accessing session state values;

receiving one or more calls to one or more second functions to facilitate construction of client applications, the one or more second functions comprising functions for creating windowing graphical user interface environments and graphical functionality;

receiving one or more calls to one or more third functions to facilitate connectivity to data sources and XML functionality, the one or more third functions comprising functions for management of data from multiple data sources and XML processing; receiving one or more calls to one or more fourth functions to access system and runtime resources, the one or more fourth functions comprising functions for programmatic access to configuration settings, application debugging and code execution tracing, customization of text according to cultural related information, synchronous and asynchronous reading from and writing to data streams and files, creation and management of various culture-specific resources, system security and permissions, installation and running of services, character encoding, and multi-threaded programming; and

using a common language runtime layer that allows seamless multi-language development, with cross language inheritance and translates Web applications written in different programming languages into an intermediate language that is supported by the common language runtime layer and configured to access services requested by the client applications, whereby a seamless integration provides a robust and secure execution environment for multiple programming languages.

n. Claims 37-40 (Canceled)

o. Claim 41;

A method implemented at least in part by a computer, for exposing resources using an application program interface, the method comprising:

exposing a first group of services related to creating Web applications, the first group of services comprising:

constructing Web services;

temporary caching resources;

performing an initial configuration;

creating controls and Web pages;

enabling security in Web server applications; and

accessing session state values;

exposing a second group of services related to constructing client applications, the second group of services comprising:

creating windowing graphical user interface environments; and

enabling graphical functionality;

exposing a third group of services related to data and handling XML documents, the third group comprising:

enabling management of data from multiple data sources; and

functions that enable XML processing;

exposing a fourth group of services related to base class libraries, the fourth group of services comprising:

defining various collections of objects;

accessing configuration settings and handling errors in configuration files;

debugging and tracing code execution;

customizing data according to cultural related information;

inputting and outputting of data;

enabling a programming interface to network protocols;

viewing loaded types, methods, and fields;

creating, storing and managing various culture-specific resources;

enabling system security and permissions;

installing and running services;

enabling character encoding;

enabling multi-threaded programming; and

facilitating runtime operations; and

providing a common language runtime layer, with cross-language inheritance that translates Web applications written in different programming languages into an intermediate language,

the intermediate language being:

executed natively by the common language runtime layer; and

configured to access resources requested by the client applications;

wherein, the different program languages are selected from a plurality of programming languages, the plurality of programming languages comprising:

Visual Basic;

C++;

C#;

COBOL;

Jscript;

Perl;

Eiffel; and

Python.

2. In the title:

Pursuant to MPEP 606.01, the title has been changed to read as follows:

-- A Computing system and method for allowing plurality of applications written in different programming languages to communicate and request resources or services via a common language runtime layer --;

3. In the specification:

- page 1, line 23, replace "U.S. Patent No. ----------, entitled "Application Program Interface for Network Software Platform", which issued ---------- from application Serial No. 09/902,812, filed July 10, 2001." with -- Application Serial No. 09/902,812, filed July 10, 2001, entitled Application Program Interface for Network Software Platform. --;
- page 2, line 3, replace "U.S. Patent No. ----------, entitled "Application Program Interface for Network Software Platform", which issued ---------- from application Serial No. 09/901,555, filed July 10, 2001." with -- Application Serial No. 09/901,555 (Now Abandoned), filed July 10, 2001, entitled Application Program Interface for Network Software Platform. --;

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHARLES E. ANYA whose telephone number is (571)272-3757. The examiner can normally be reached on 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

cea.

/Li B. Zhen/
Primary Examiner, Art Unit 2194